

JETri, Vol. 15, No. 1, Agustus 2017, Hlm. 81 -100, P-ISSN 1412-0372, E-ISSN 2541-089X

FLOWER POLLINATION ALGORITHM **UNTUK OPTIMASI PENGENDALI PID** **PADA PENGENDALIAN KECEPATAN** **MOTOR INDUKSI**

M.Ruswandi Djalal¹, Yusuf Yunus¹, Andi Imran² dan Herlambang Setiadi³

¹Jurusan Teknik Mesin, Politeknik Negeri Ujung Pandang

Jalan Perintis Kemerdekaan km 10 Tamalanrea, Makassar 90245

²Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember

Jalan Raya ITS, Keputih, Sukolilo, Surabaya, Jawa Timur 60111

³ School of Electrical Engineering, The University of Queensland

Brisbane QLD 4072 Australia

E-mail: wandi@poliupg.ac.id

ABSTRACT

The use of Proportional Integral Derivative (PID) controller in induction motors is becoming more and more popular, because of its simple structure. PID controller requires proper parameter setting for optimal performance on the induction motor. The most commonly used method is by trial and error to determine parameters of the PID controller, but the results obtained are not optimal and incorrect PID controller's parameters will damage the system. For that reason, in this research it will be shown one of PID parameters tuning method by using Flower Pollination Algorithm (FPA) to optimize and determine the exact parameters of the PID. FPA is a method that is being adapted and applied as a smart algorithm to solve optimization problem. The PID parameters tuning in this study gives results that the value of k_p , k_i and k_d are 0.4213, 0.2337 and 0.027 respectively. As a comparison, this study has also used Firefly, Cuckoo Search, Particle Swarm, Imperialist Competitive, Ant Colony, Differential Evolution, and Bat method. The FPA method can well tune the PID parameters, so that the resulting overshoot is very small in comparison with the other methods, it is at 1,019 from the set point. Compared with other methods, the settling time is also very fast, that is 0.3second.

Keywords: PID, FPA, Bee-Colony, Cuckoo, Firefly

ABSTRAK

Penggunaan pengendali Proportional Integral Derivative (PID) pada motor induksi menjadi semakin populer, karena strukturnya yang sederhana. Pengendali PID memerlukan pengaturan parameter yang tepat untuk kinerja optimal pada motor induksi. Metode yang paling umum digunakan adalah dengan metode trial and error untuk menentukan parameter pengendali PID, namun hasil yang didapat tidak optimal dan parameter pengendali PID yang tidak tepat akan merusak sistem. Oleh karena itu, dalam penelitian ini, diperlihatkan salah satu metode penalaan parameter PID dengan menggunakan metode Flower Pollination Algorithm (FPA) untuk mengoptimalkan dan menentukan parameter PID yang tepat. FPA adalah salah satu metode yang diadaptasi dan diterapkan sebagai algoritma cerdas untuk mengatasi masalah optimasi. Hasil penalaan yang diperoleh adalah nilai k_p , k_i , dan k_d masing-masing sebesar 0,4213, 0,2337, dan 0,0274. Sebagai perbandingan, penelitian ini juga menggunakan metode Firefly, Cuckoo Search, Particle Swarm, Imperialist Competitive, Ant Colony, Differential Evolution, dan metode Bat. Metode FPA dapat menala parameter PID sehingga overshoot yang dihasilkan sangat kecil dibandingkan dengan metode lainnya yaitu sebesar 1,019 terhadap set point. Waktu settling yang diperoleh juga sangat cepat dibandingkan dengan metode lainnya, yaitu 0,3 detik.

Kata kunci: PID, FPA, Bee-Colony, Cuckoo, Firefly

1. PENDAHULUAN

Motor induksi merupakan motor listrik AC yang paling banyak digunakan pada sektor industri baik dalam skala kecil, menengah maupun besar. Motor induksi yang digunakan adalah jenis rotor sangkar tupai. Motor induksi memiliki banyak keunggulan seperti konstruksi yang sangat kuat dan sederhana, harga relatif murah, ketahanan dan keandalan tinggi, efisiensi tinggi, dan biaya pemeliharaan yang rendah selama bertahun-tahun.

Pada motor induksi terdapat beberapa parameter nonlinier, salah satunya tahanan rotor yang memiliki nilai yang bervariasi untuk kondisi operasi yang berbeda. Hal ini menyebabkan kendali kecepatan motor induksi menjadi kompleks dan lebih mahal daripada kendali motor DC. Untuk mendapatkan kecepatan konstan dan meningkatkan kinerja motor induksi maka dibutuhkan suatu kendali. Salah satu metode kendali yang digunakan untuk mengendalikan kecepatan pada motor induksi menggunakan pengendali *Proportional Integral Derivative* (PID). Pengendali PID memiliki kinerja yang baik sehingga banyak digunakan dalam dunia industri. Namun memiliki kelemahan yaitu membutuhkan perhitungan matematik yang rumit.

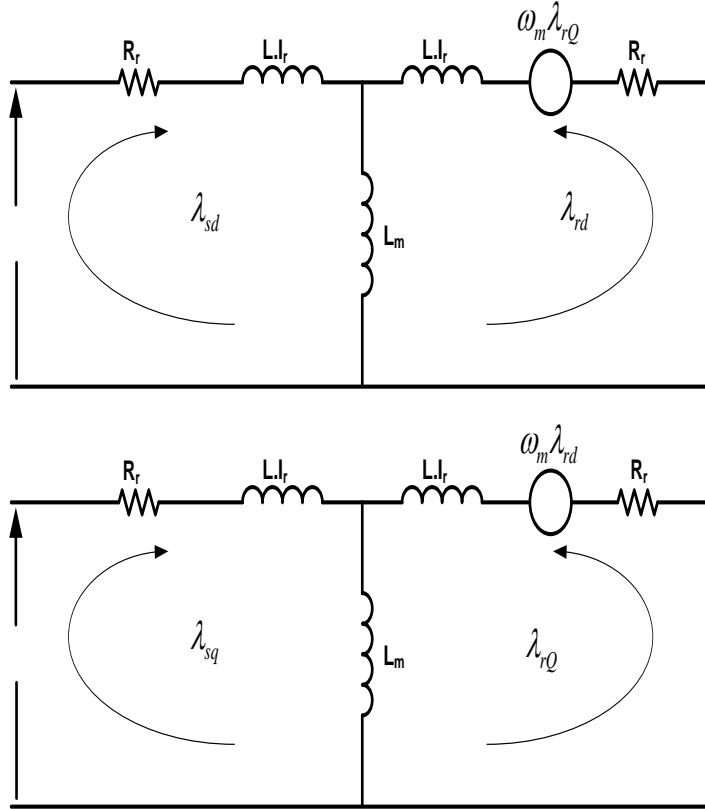
Beberapa metode optimasi berbasis metode konvensional maupun metode cerdas telah banyak digunakan untuk mengoptimasi penentuan parameter PID, diantaranya dengan metode konvensional [1,2], yang kemudian berkembang dengan menggunakan metode cerdas seperti *Artificial Bee Colony* [3], *Particle Swarm Optimization* [4], *Genetic Algorithm* [5], *Bacterial Foraging* [6], *Fuzzy Logic* [7], *Imperialist Algorithm* [8], *Bat Algorithm* [9], *Firefly* [10], *Ant Colony* [11], *Differential Evolution* [12], dan *Cuckoo* [13]. Beberapa penelitian di atas menunjukkan hasil yang baik dalam optimasi parameter PID.

Flower Pollination Algorithm merupakan algoritma yang ditemukan oleh Xin She Yang pada tahun 2012 [14], namun aplikasi algoritma ini masih sedikit digunakan dalam masalah optimasi. Berdasarkan [14], algoritma ini sangat cepat dalam melakukan komputasi dan merupakan pengembangan dari algoritma cerdas sebelumnya yang juga ditemukan oleh Xin She Yang.

Untuk mengatasi permasalahan di atas, maka pada penelitian ini diusulkan pemanfaatan metode *Flower Pollination Algorithm* untuk mengendalikan kecepatan motor induksi tiga fasa. Sebagai pembanding kinerja digunakan pula beberapa metode yaitu metode *Firefly*, *Cuckoo Search*, *Particle Swarm*, *Imperialist Competitive*, *Ant Colony*, *Differential Evolution*, dan *Bat algorithm*.

2. KAJIAN PUSTAKA

Berdasarkan [15], sebuah model yang mendeskripsikan motor induksi sebagai sebuah sistem yang terdiri dari vektor-vektor yang didefinisikan dengan dua koordinat ortogonal dan polar. Komponen vektor ruang diasosiasikan dengan pasangan variabel, yaitu bagian *direct* dan *quadrature* (d-q). Gambar 1 memperlihatkan rangkaian ekuivalen dari sebuah motor induksi. Motor induksi sebagai sistem yang memiliki banyak masukan, dalam hal ini model (d-q) *reference frame* adalah $V_{sd}(t)$, $V_{sq}(t)$ dan $T_L(t)$ memiliki karakteristik untuk mendapatkan fungsi alih yang lebih rumit jika dibandingkan dengan sistem masukan tunggal. Untuk mendapatkan fungsi alih, maka salah satu dari tegangan acuan masukan dan $T_L(t)$ harus sama dengan nol [15].



Gambar 1 Rangkaian Ekuivalen Motor Induksi [15]

$$(J_{eq}s + B_m)\omega_m(t) = T_{em}(t) - T_L(t) \quad (1)$$

$$\omega_m(s) = \frac{T_{em}(s)}{(J_{eq}s + B_m)} = \frac{n_p \cdot L_m (\lambda_{rd} \cdot i_{sq}(s) - \lambda_{rq} \cdot i_{sd}(s))}{L_r \cdot (J_{eq}s + B_m)} \quad (2)$$

Dari [15] diperoleh :

$$V_{sd}(t) = n_p \cdot \beta \cdot \omega_m(t) \lambda_{rq} + \eta \cdot \beta \cdot \lambda_{rd} + \gamma \cdot i_{sd}(t) + \frac{d}{dt} i_{sd}(t) \cdot \sigma \cdot L_s$$

Jika β , η , dan γ disubstitusikan :

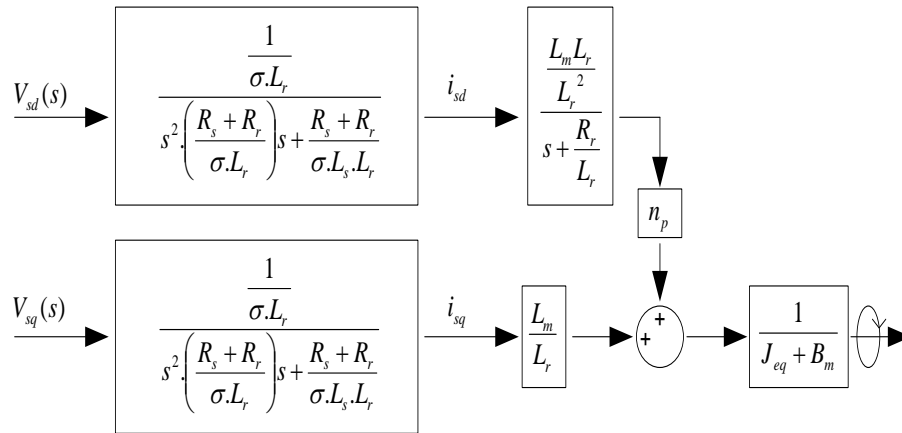
$$V_{sd}(t) = ((n_p \cdot \omega_m(t) \lambda_{rq} + \frac{R_r}{L_r} \lambda_{rd}) \cdot \frac{L_m}{\sigma \cdot L_s \cdot L_r} + (\frac{L_m}{\sigma \cdot L_s \cdot L_r^2} + \frac{R_s}{\sigma \cdot L_s}) i_{sd}(t) + \frac{d}{dt} i_{sd}(t) \cdot \sigma \cdot L_s$$

$$V_{sd}(s) = n_p \cdot \frac{L_m}{L_r} \omega_m(s) \lambda_{rq} + \frac{R_r L_m}{L_r^2} \lambda_{rd} + (\frac{L_m}{L_r} + R_s + \sigma \cdot L_s \cdot s) i_{sd}(s) \quad (3)$$

Maka persamaan fungsi alih motor induksi adalah :

$$\frac{\omega_m(s)}{V_{sd}(s)} = \frac{\frac{n_p \cdot L_m (\lambda_{rd} i_{sq}(s) - \lambda_{rq} i_{sd}(s))}{L_r \cdot (J_{eq} s + B_m)}}{\frac{n_p \cdot \frac{L_m}{L_r} \omega_m(s) \lambda_{rq} + \frac{R_r L_m}{L_r^2} \lambda_{rd} + (\frac{L_m}{L_r^2} + R_s + \sigma L_s s) i_{sd}(s)}} = \frac{\frac{n_p \cdot L_m (\lambda_{rd} i_{sq}(s) - \lambda_{rq} i_{sd}(s))}{L_r}}{J_{eq} \sigma L_s s^2 i_{sd}(s) + ((R_s + \frac{L_m}{L_r^2}) i_{sd}(s) + \frac{R_r L_m}{L_r^2} \lambda_{rd}) J_{eq} s + \frac{n_p^2 L_m^2}{L_r^2} (\lambda_{rd} i_{sq}(s) - \lambda_{rq} i_{sd}(s))} \quad (4)$$

Dari fungsi alih pada persamaan (4) diperoleh diagram blok motor induksi seperti diperlihatkan pada Gambar 2.



Gambar 2 Diagram Blok Motor Induksi [15]

3. PERANCANGAN

3.1 Pemodelan Motor Induksi

Jenis motor induksi yang akan digunakan pada penelitian ini adalah motor jenis *squirrel cage* (sangkar tupai). Agar dapat digunakan dalam pengendalian maka harus diketahui terlebih dahulu nilai dari parameter-parameter yang dimiliki oleh motor induksi tersebut. Spesifikasi fisis motor induksi adalah seperti yang ditunjukkan pada Tabel 1 berikut ini.

Tabel 1 Spesifikasi Fisis Motor Induksi *Squirrel Cage* [15]

Simbol	Nama	Nilai	Satuan
n	Tegangan sumber terukur	230	<i>Volt.</i>
P	Jumlah pole	2	-
f_s	Frekuensi stator	60	<i>Hz</i>
J_{eq}	Momen inersia stator	0.3	<i>Kg.m²</i>
R_s	Tahanan stator	0.08	<i>Ohm</i>
R_r	Tahanan rotor	0.04	<i>Ohm</i>
L_s	Induktansi stator	15.4	<i>mH</i>
L_r	Induktansi rotor	16.38	<i>mH</i>
L_m	Induktansi magnetik	14.60	<i>mH</i>
S	Slip	0.02	-
B_m	Koefisien gesekan	-	<i>Nm/rad/sec</i>
k_p	Koefisien penguat proporsional	-	-
k_i	Koefisien penguat integral	-	-
k_d	Koefisien penguat diferensial	-	-

Nilai-nilai dari parameter seperti yang ditunjukkan pada Tabel 1 ini akan disubstitusikan ke dalam bentuk sebuah fungsi alih dari persamaan di atas sebagai fungsi alih motor induksi yang akan dilengkapi dan disempurnakan spesifikasi keluarannya dengan pengendali.

$$\frac{\omega_m(s)}{V_{sd}(s)} = \frac{1.78}{0.72 \times 10^{-3} s^2 + 0.0157s + 3.168} \quad (5)$$

$$\frac{\omega_m(s)}{V_{sd}(s)} = \frac{2470}{s^2 + 21.79s + 4400} \quad (6)$$

3.2 Pemodelan Pengendali PID [15,16]

Pengendali PID adalah salah satu pengendali yang sudah banyak digunakan pada aplikasi industri karena strukturnya yang sederhana [15,16].

$$u(t) = k_p \left[e(t) + \frac{1}{k_i} \int_0^t e(t) dt + k_d \frac{de_t}{dt} \right] \quad (7)$$

$u(t)$: nilai kendali yang dihitung oleh pengendali PID

k_p : koefisien penguat proporsional

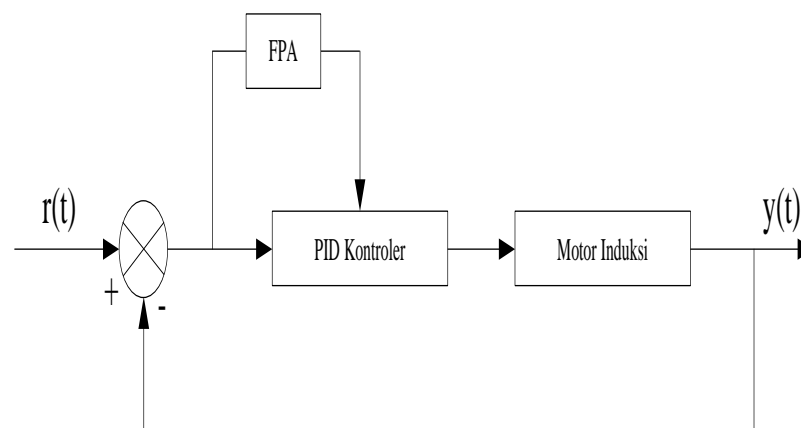
k_i : koefisien penguat integral

k_d : koefisien penguat diferensial.

Fungsi dari ketiga elemen tersebut adalah:

- koefisien penguat proporsional k_p : menggambarkan deviasi sinyal $e(t)$ dari sistem kendali proporsional, ketika sinyal $e(t)$ ada, pengendali PID segera menghasilkan efek kendali untuk mengurangi penyimpangan;
- koefisien penguat integral k_i : digunakan untuk menghilangkan kesalahan statis dan meningkatkan stabilitas sistem;
- koefisien penguat diferensial k_d : mencerminkan perubahan penyimpangan sinyal, memperkenalkan sinyal koreksi sebelum penyimpangan nilai sinyal menjadi lebih besar dan dapat mempercepat respon sistem untuk mengurangi pengaturan waktu transien.

Oleh karena itu, hal yang terutama dalam merancang pengendali PID adalah menentukan ketiga parameter, serta bagaimana menghitung nilai ketiga parameter PID tersebut yaitu k_p , k_i dan k_d . Dalam penelitian ini, FPA digunakan untuk mencari parameter optimal dari PID. Blok diagram sistem kendali yang dibangun adalah seperti ditunjukkan pada Gambar 3, yang terdiri dari input, proses dan *output*. FPA akan melakukan optimasi parameter pengendali PID, kemudian *output* kecepatan akan diumpanbalikkan ke sisi input untuk dijumlahkan dengan *set point*.



Gambar 3 Blok Diagram Sistem Kendali Kecepatan Motor Induksi dengan Implementasi *Flower Pollination Algorithm* untuk Optimasi Pengendali PID

3.3 Flower Pollination Algorithm

3.3.1 Konsep Flower Pollination Algorithm

Aplikasi dalam bidang teknik dan industri seringkali melibatkan masalah optimasi dengan kendala-kendala yang kompleks. Masalah tersebut acapkali bersifat nonlinier, sehingga masalah ini sulit untuk dapat diselesaikan. Metode-metode klasik seringkali tidak dapat bekerja dengan baik untuk menyelesaikan masalah-masalah yang nonlinier dan *multimodal* seperti ini. Salah satu penyebabnya adalah karena di dalam metode klasik seperti metode Newton dibutuhkan informasi gradien dan di dalam metode lainnya, dibutuhkan Hessian, padahal tidak semua fungsi memiliki gradien di daerah definisinya. Penyebab lainnya adalah kekonvergenan metode klasik tersebut biasanya bergantung pada nilai titik tebakan awal, sehingga dalam proses iterasinya, ada peluang terjadinya proses pencarian terjebak di nilai optimum lokal.

Flower Pollination Algorithm merupakan salah satu algoritma *metaheuristic* yang dapat menggantikan metode klasik tersebut. Beberapa fenomena penyerbukan bunga adalah sebagai berikut:

- a. penyerbukan biotik dan penyerbukan silang dipandang sebagai penyerbukan global dengan organisme penyerbuk pembawa serbuk sari yang melakukan gerakan *Lévy Flights*;
- b. penyerbukan abiotik dan penyerbukan sendiri dipandang sebagai penyerbukan lokal;
- c. *flower constancy* dipandang sebagai peluang reproduksi yang proporsional dengan kesamaan dari dua bunga yang terlibat;
- d. penyerbukan lokal dan global diatur oleh peluang berpindah..

Terdapat dua langkah kunci dalam algoritma ini, yaitu penyerbukan global dan penyerbukan lokal. Pada langkah penyerbukan global, serbuk sari dari bunga dibawa oleh hewan penyerbuk seperti serangga, dan serbuk sari dapat melakukan perjalanan jarak jauh karena serangga dapat terbang dan bergerak di daerah yang luas. Proses ini bisa menghasilkan solusi yang paling optimal yang direpresentasikan dengan aturan pertama, ditambah dengan fenomena *flower constancy* dapat direpresentasikan secara matematis sebagai:

$$x_i^{t+1} = x_i^t + \gamma L(\lambda) (x_i^t - g^*) \quad (8)$$

x_i^t : serbuk sari i atau vektor solusi x_i pada iterasi t ,

g^* : solusi terbaik diiterasi yang sedang berlangsung.

L : kekuatan dari penyerbukan, yang pada dasarnya merupakan besar langkah.

Pada Tabel 2 berikut ini diperlihatkan analogi penyerbukan bunga dengan optimasi.

Masalah Optimasi:

maks atau min $f(x_i)$, $x_i = (x_1, x_2, \dots, x_n) \in S^d \subset R^d, i = 1, 2, \dots, n$

Tabel 2 Analogi Penyerbukan Bunga dengan Optimasi [16]

Penyerbukan Bunga	Optimasi
Fenomena Penyerbukan	Permasalahan Optimasi
Daerah Cakupan Penyerbukan	Interval nilai $S^d, x \in S^d$
Populasi Bunga	Kandidat Solusi (x_1, x_2, \dots, x_n)
Tingkat kecocokan Bunga	Fungsi objektif f
Kejadian usaha penyerbukan	Iterasi
Gerak Penyerbukan Bunga	Pencarian Solusi pada Daerah Definisi
Dipilihnya Individu yang cocok	Keadaan Optimum

Berikut ini disajikan *Pseudo Code* dari FPA

Objektif min atau maks $f(x)$, $x = (x_1, x_2, \dots, x_d)$

Inisiasi populasi n bunga (gamet serbuksari) sebagai solusi acak

Temukan solusi terbaik g^* di populasi awal

Definisikan *switch probability* $p \in [0,1]$

Definisikan kriteria berhenti (akurasi atau iterasi maksimum)

While ($t < \text{Iterasi Maksimum}$)

For $i=1:n$ (seluruh n bunga dari populasi)

If $\text{rand} < p$,

Ambil vektor langkah L berdimensi d yang memenuhi distribusi Levy

Penyerbukan global $x_i^{t+1} = x_i^t + \gamma L(\lambda) (x_i^t - g^*)$

```

Else
    Ambil  $\epsilon$  dari distribusi uniform (0,1)
    Lakukan penyerbukan lokal  $x_i^{t+1} = x_i^t + \epsilon (x_j^t - x_k^t)$ 
    End if
    Evaluasi solusi baru
    Jika solusi yang baru lebih baik, perbarui solusi ini ke dalam populasi
    End for
    Temukan solusi terbaik  $g^*$ 
    End while
    Tampilkan solusi terbaik yang ditemukan
    
```

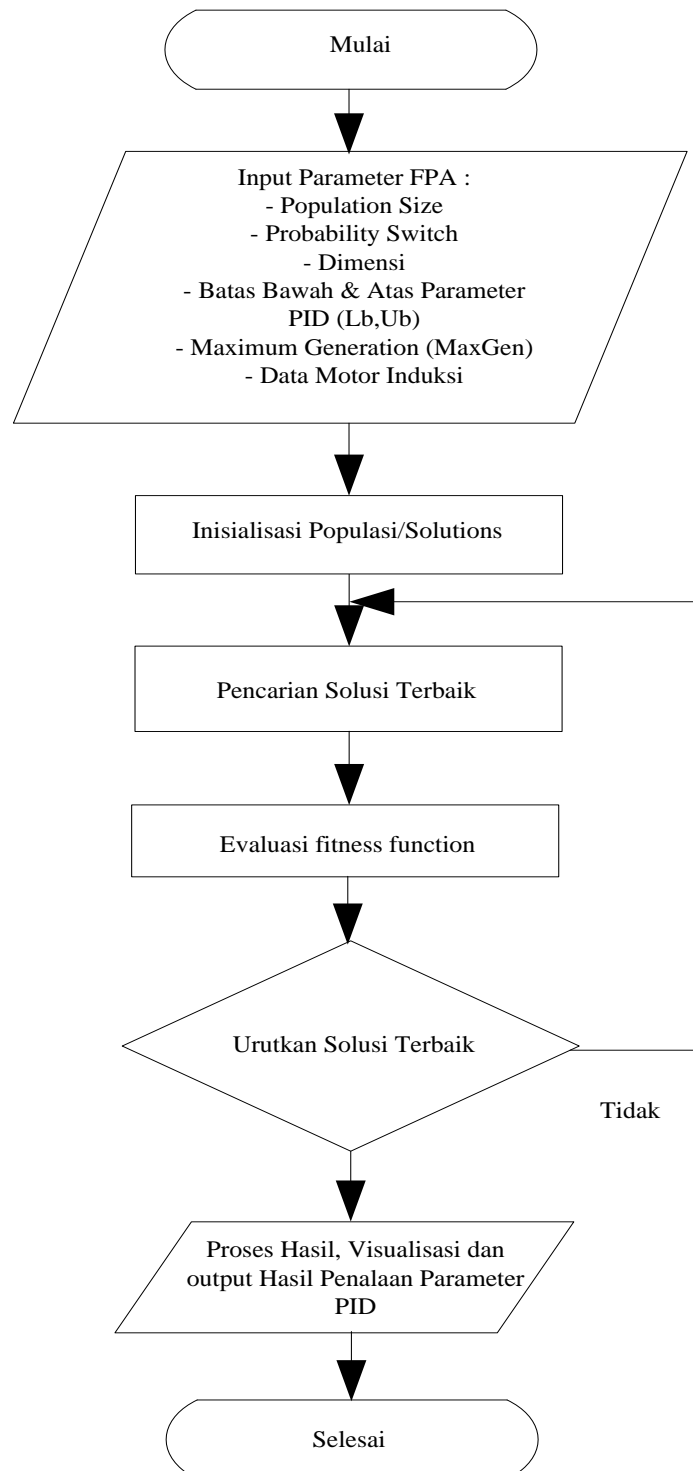
3.3.2 Penalaan PID dengan *Flower Pollination Algorithm*

Gambar 4 menunjukkan diagram alir FPA yang diimplementasikan pada penelitian ini untuk menala parameter-parameter dari PID. Fungsi objektif yang dipergunakan untuk menguji kestabilan sistem adalah dengan *Integral Time Absolut Error* (ITAE) [16]. Fungsi obyektif akan mencari parameter yang tepat dengan meminimalkan *error* kecepatan ω .

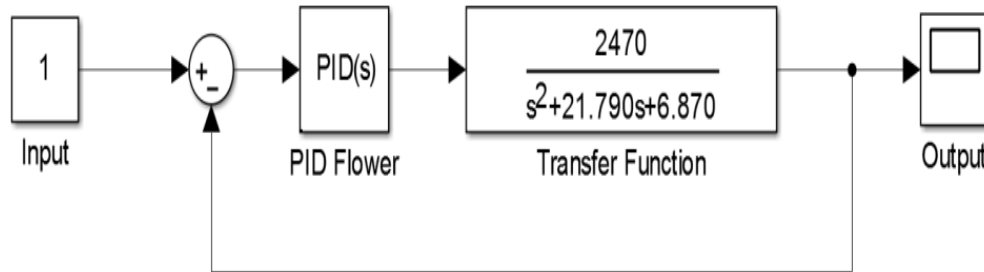
$$ITAE = \int_0^t t |\Delta\omega(t)| dt \quad (9)$$

Parameter PID yang ditala oleh FPA adalah k_p , k_i dan k_d . Adapun untuk diagram alir proses penalaan parameter PID dengan menggunakan metode FPA ditunjukkan pada *flowchart* pada Gambar 4 dan Gambar 5 menunjukkan pemodelan motor induksi pada Simulink Matlab 2015, tanpa kendali, dengan *Firefly*, *Cuckoo Search*, *Particle Swarm*, *Imperialist Competitive*, *Ant Colony*, *Diferential Evolution*, dan *Bat algorithm*.

Berdasarkan persamaan (6), dapat dibuat pemodelan untuk motor induksi dengan Simulink seperti yang ditunjukkan pada Gambar 5.



Gambar 4 *Flowchart* Optimasi FPA



Gambar 5 Pemodelan Motor Induksi di Simulink

4. HASIL DAN PEMBAHASAN

Pada penelitian ini akan digunakan beberapa metode cerdas yang digunakan untuk mencari parameter optimal dari pengendali PID dengan menggunakan persamaan 5 sebagai fungsi tujuan (*Objective Function*). Metode yang digunakan adalah sebagai berikut:

- firefly algorithm*, yaitu algoritma yang diadopsi dari perilaku kunang-kunang yang berkedip di kegelapan;
- cuckoo search*, yaitu algoritma yang diadopsi dari perilaku burung cuckoo dalam mencari sarang terbaik;
- particle swarm*, yaitu algoritma yang diadopsi dari perilaku kawanan partikel dalam mencari makanan;
- ant colony*, yaitu algoritma yang diadopsi dari perilaku semut dalam mencari sumber makanan secara cepat;
- flower pollination algorithm*, yaitu algoritma yang diadopsi dari proses penyerbukan bunga.

4.1 Penalaan Pengendali PID Menggunakan *Firefly Algorithm* (FA)

Jumlah kunang-kunang (*firefly*) diasumsikan sebagai solusi optimal dari nilai PID yang dicari. Korelasi algoritma FA dan *problem* adalah seperti berikut:

- inisialisasi populasi *firefly* [k_p , k_i , k_d];
- proses (pindah ke *firefly* yg lebih terang);
- pilih *best firefly*, k_p “best”, k_i “best”, k_d “best”.

4.2 Penalaan Pengendali PID Menggunakan *Cuckoo Search Algorithm* (CSA)

Jumlah sarang (*nest*) diasumsikan sebagai solusi optimal dari nilai parameter PID yang dicari. Korelasi algoritma CSA dan *problem* adalah seperti berikut:

1. inisialisasi sarang [k_p , k_i , k_d];
2. proses (pencarian sarang *via Levy Flight*);
3. pilih *best nest*, k_p "best", k_i "best", k_d "best".

4.3 Penalaan Pengendali PID Menggunakan *Particle Swarm Optimization Algorithm* (PSO)

Posisi terbaik partikel (*swarm*) diasumsikan sebagai solusi optimal dari nilai parameter PID yang dicari. Korelasi algoritma PSO dan *problem* adalah seperti berikut:

1. inisialisasi partikel [k_p , k_i , k_d];
2. *random* (proses);
3. pilih *best position*, k_p "best", k_i "best", k_d "best".

4.4 Penalaan Pengendali PID Menggunakan *Ant Colony Optimization* (ACO)

Sumber makanan semut (*ant*) diasumsikan sebagai solusi optimal dari nilai parameter PID yang dicari. Korelasi algoritma ACO dan *problem* adalah seperti berikut:

1. inisialisasi populasi semut (k_p , k_i , k_d);
2. proses (*tour*);
3. pilih *best route*, k_p "best", k_i "best", k_d "best".

4.5 Penalaan Pengendali PID Menggunakan *Differential Evolution* (DE)

Populasi baru diasumsikan sebagai solusi optimal dari nilai parameter PID yang dicari. Korelasi algoritma DE dan *problem* adalah seperti berikut:

1. inisialisasi populasi (k_p , k_i , k_d);
2. mutasi (populasi);
3. populasi baru, k_p "best", k_i "best", k_d "best".

4.6 Penalaan Pengendali PID Menggunakan *Bat Algorithm* (BA)

Sumber makanan diasumsikan sebagai solusi optimal dari nilai parameter PID yang dicari. Korelasi algoritma BA dan *problem* adalah seperti berikut:

1. inialisasi populasi *bat* (k_p , k_i , k_d);
2. proses ekolokasi;
3. *best food*, k_p “*best*”, k_i “*best*”, k_d “*best*”.

4.7 Penalaan Pengendali PID Menggunakan *Imperialist Competitive Algorithm* (ICA)

Kerajaan (*empire*) terbaik baru diasumsikan sebagai solusi optimal dari nilai parameter PID yang dicari. Korelasi algoritma ICA dan *problem* adalah seperti berikut:

1. inialisasi *empire* (k_p , k_i , k_d);
2. kompetisi (proses);
3. *best empire*, k_p “*best*”, k_i “*best*”, k_d “*best*”.

4.8 Penalaan Pengendali PID Menggunakan *Flower Pollination Algorithm* (FPA)

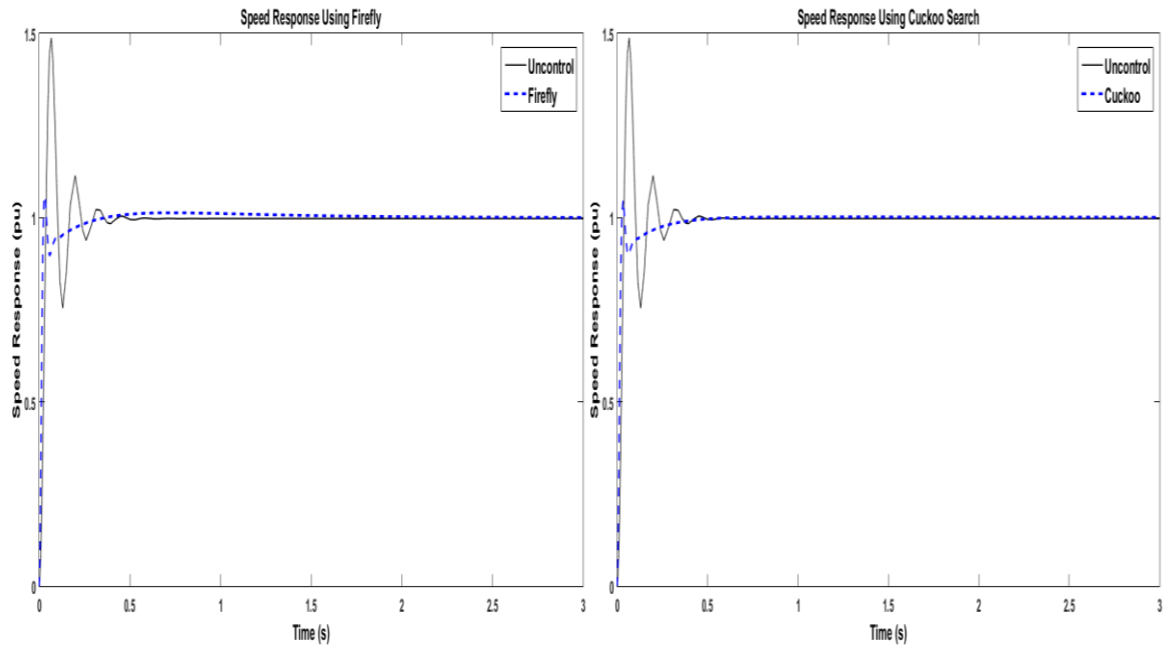
Populasi baru diasumsikan sebagai solusi optimal dari nilai parameter PID yang dicari. Korelasi algoritma FPA dan *problem* adalah seperti berikut:

1. inialisasi populasi *flower* (k_p , k_i , k_d);
2. proses (*Foraging*);
3. *best solution*, k_p “*best*”, k_i “*best*”, k_d “*best*”.

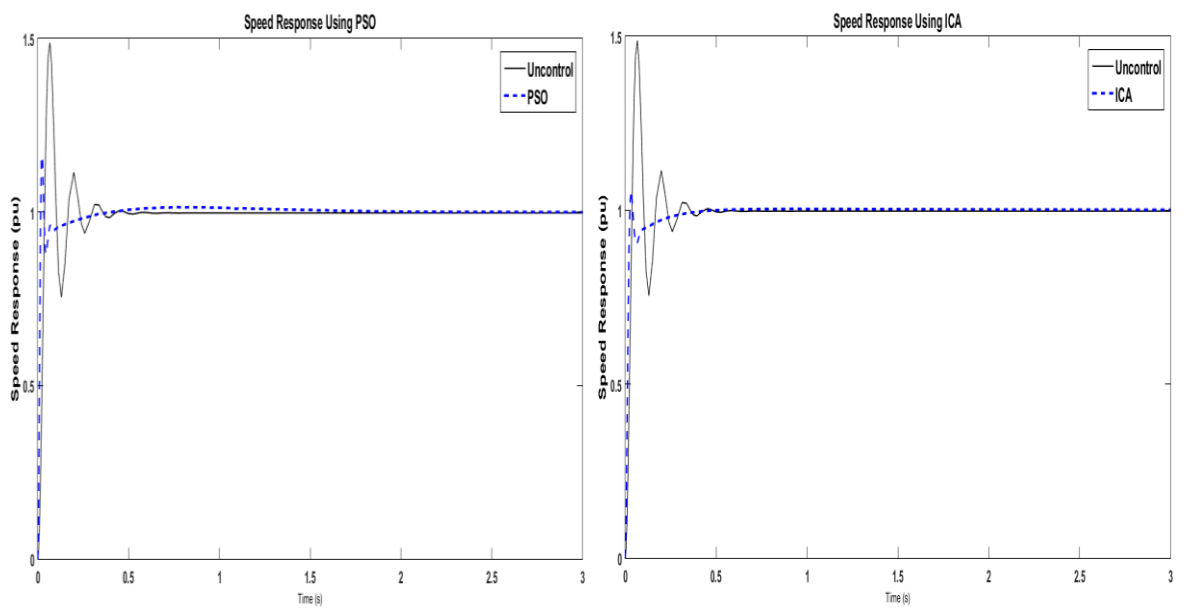
Tabel 3 Hasil Optimasi Nilai Parameter PID Menggunakan Metode Cerdas

Param.	FA	CSA	PSO	ACO	DE	FPA	ICA	BA
k_p	0.3392	0.3388	0.3889	0.3359	0.4337	0.4213	0.3654	0.2960
k_i	0.4150	0.1398	0.6355	0.3173	0.3919	0.2337	0.1922	0.0923
k_d	0.0500	0.0449	0.0825	0.0427	0.0407	0.0274	0.0453	0.0491

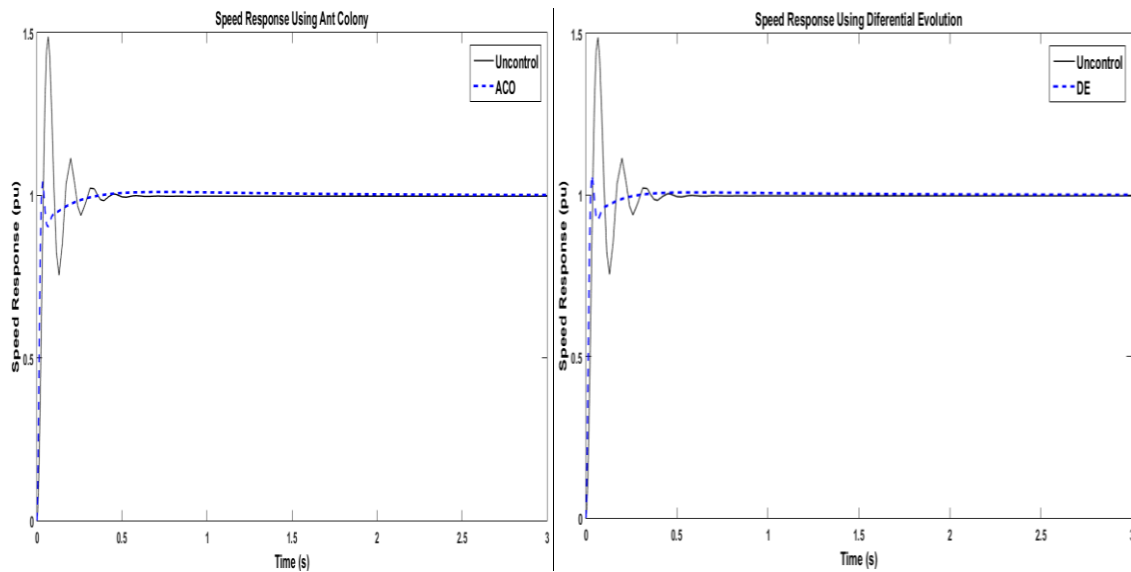
Gambar berikut menunjukkan respon kecepatan untuk masing-masing model pengontrolan.



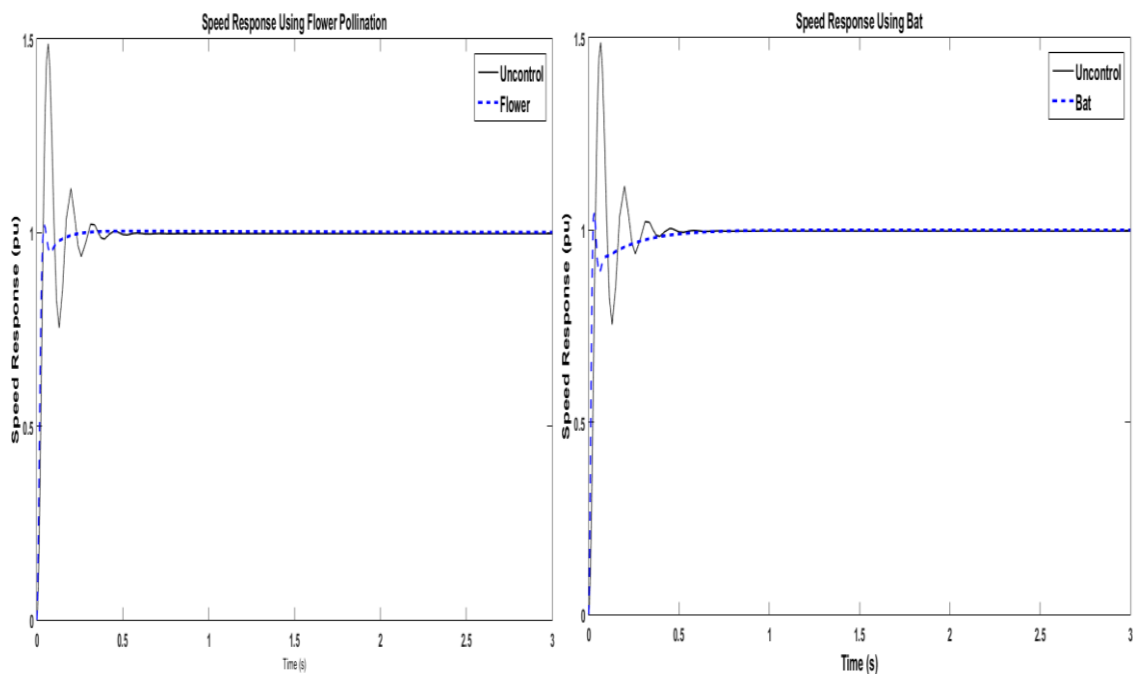
Gambar 6 Respon Kecepatan Menggunakan Metode PID - FA dan PID - CSA



Gambar 7 Respon Kecepatan Menggunakan Metode PID - PSO dan PID - ICA



Gambar 8 Respon Kecepatan Menggunakan Metode PID – ACO dan PID - DE



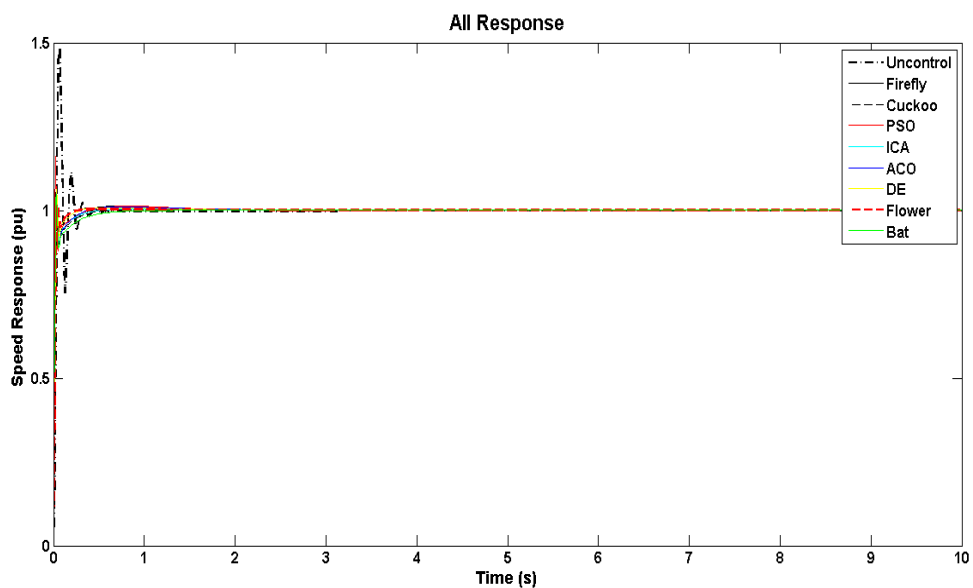
Gambar 9 Respon Kecepatan Menggunakan Metode PID - FPA dan PID - BA

Gambar 10 menunjukkan hasil respon kecepatan motor dengan beberapa model pengontrolan dan Tabel 4 menunjukkan nilai *overshoot* untuk masing-masing metode optimasi.

Tabel 4 *Overshoot* Masing-masing Metode Optimasi

<i>Method</i>	<i>Overshoot (pu)</i>	<i>Settling Time (s)</i>
<i>Uncontrol</i>	1.488	0.6
<i>Firefly</i>	1.059	1.8
<i>Cuckoo Search</i>	1.048	0.45
<i>Particle Swarm</i>	1.161	1.9
<i>Imperialist Competitive</i>	1.056	0.5
<i>Ant Colony</i>	1.042	1.4
<i>Diferential Evolution</i>	1.062	1.6
<i>Flower Pollination</i>	1.019	0.3
<i>Bat</i>	1.047	0.8

Dari tabel tersebut dapat dilihat bahwa, dengan menggunakan metode FPA respon *overshoot* dapat diredam, sehingga *overshoot* yang dihasilkan menjadi sangat kecil yaitu sebesar 1.019. Gambar 10 memperlihatkan plot semua model pengontrolan, dimana FPA menunjukkan hasil yang sangat baik. Dapat disimpulkan, penggunaan FPA sebagai metode optimasi PID dapat mencari nilai optimal dari PID dengan baik.



Gambar 10 Respon Kecepatan Menggunakan Masing-masing Metode

5. KESIMPULAN

1. Dengan menggunakan metode cerdas FPA sebagai metode penalaan pengendali PID, didapatkan hasil penalaan nilai parameter PID yang optimal yaitu nilai k_p sebesar 0.4213, k_i sebesar 0.2337 dan k_d sebesar 0.0274.
2. Dari hasil simulasi dapat disimpulkan bahwa respon kecepatan motor induksi dengan pengendali PID-FPA, didapatkan *overshoot* terkecil yaitu 1.019 pu, dan *settling time* yang sangat cepat yaitu 0.3sekon dibandingkan dengan metode-metode lain di mana sistem sudah berada pada kondisi *steady* sebelum $t = 1$ sekon.
3. Metode FPA dapat digunakan untuk mengoptimasi parameter PID, hasil yang didapat untuk kinerja PID sangat baik karena menunjukkan respon yang cepat untuk kendali motor induksi.

DAFTAR PUSTAKA

- [1] P. M. Meshram dan R. G. Kanojiya. "Tuning of PID Controller Using Ziegler-Nichols Method for Speed Control of DC Motor." dalam *IEEE-International Conference on Advances in Engineering, Science and Management (ICAESM-2012)*, 2012, hlm. 117-122.
- [2] J. R. Manepalli dan C. Raja. "Speed Control of Induction Motor by ZN Method and Genetic Algorithm Optimization with PI and PID Controller."
- [3] A. G. Abro dan J. M. Saleh. "Multiple-Global-Best Guided Artificial Bee Colony Algorithm for Induction Motor Parameter Estimation," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 22, hlm. 620-636, 2014.
- [4] D. H. Kim. "GA-PSO Based Vector Control of Indirect Three Phase Induction Motor." *Applied Soft Computing*, vol. 7, hlm. 601-611, 2007.
- [5] S. V. Ustun dan M. Demirtas. "Modeling and Control of V/f Controlled Induction Motor Using Genetic-ANFIS Algorithm." *Energy Conversion and Management*, vol. 50, hlm. 786-791, 2009.

- [6] D. H. Kim. "Robust Tuning of Embedded Intelligent PID Controller for Induction Motor Using Bacterial Foraging Based Optimization." dalam *International Conference on Embedded Software and Systems*, 2004, hlm. 137-142.
- [7] S. M. Kim dan W. Y. Han. "Induction Motor Servo Drive Using Robust PID-Like Neuro-Fuzzy Controller." *Control Engineering Practice*, vol. 14, hlm. 481-487, 2006.
- [8] E. S. Ali. "Speed Control of Induction Motor Supplied by Wind Turbine via Imperialist Competitive Algorithm." *Energy*, vol. 89, hlm. 593-600, 2015.
- [9] K. Premkumar dan B. Manikandan. "Speed Control of Brushless DC Motor Using Bat Algorithm Optimized Adaptive Neuro-Fuzzy Inference System." *Applied Soft Computing*, vol. 32, hlm. 403-419, 2015.
- [10] E. Ali. "Speed Control of DC Series Motor Supplied by Photovoltaic System via Firefly Algorithm." *Neural Computing and Applications*, vol. 26, hlm.. 1321-1332, 2015.
- [11] C. Chengzhi, G. Xiaofeng, dan L. Yang. "Research on Ant Colony Neural Network PID Controller and Application." dalam *Eighth ACIS International Conference on, Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, 2007. *SNPD 2007*, hlm. 253-258.
- [12] L. dos Santos Coelho dan D. L. de Andrade Bernert. "A Modified Ant Colony Optimization Algorithm Based on Differential Evolution for Chaotic Synchronization." *Expert Systems with Applications*, vol. 37, hlm. 4198-4203, 2010.
- [13] M. R. Djalal, D. Ajiatmo, S. Soedibyo, dan I. Robandi. "Frequency Control Pltmh dengan Capacitive Energy Storage Menggunakan Cuckoo Search Algorithm." *SENTIA 2015*, vol. 7, 2015.
- [14] X.-S. Yang. "Flower Pollination Algorithm for Global Optimization." in *International Conference on Unconventional Computing and Natural Computation*, 2012, hlm. 240-249.

- [15] Zulfatman. "Desain Pengendalian Kecepatan Motor Induksi 3 Phase dengan PID Kendaliler." *Jurnal Gamma Universitas Muhammadiyah Malang*, 2006.
- [16] D. Lastomo. "Simulasi Pengendali Sudut Pitch Blade pada Turbin Angin dengan Flower Pollination Algorithm (FPA) untuk Mengoptimalkan Konversi Daya Listrik." Master Thesis ITS, Surabaya, 2016.